PingAccess®

Version 3.0

SDK Developer's Guide

Copyright

PingAccess

Version 3.0
July, 2014

Ping Identity Corporation
1001 17th Street, Suite 100
Denver, CO 80202
U.S.A.

Trademark

Disclaimer

Document Lifetime

Ping Identity may occasionally update online documentation between releases of the related software. Consequently, if this PDF was not downloaded recently, it may not contain the most up-to-date information. Please refer to the online documentation at *documentation.pingidentity.com* for the most current information.

From the web site, you may also download and refresh this PDF if it has been updated, as indicated by a change on this date: **July, 2014**.

SDK Developer's Guide

PingAccess SDK Developer's Guide

Preface

Preface

This document provides technical guidance for using the PingAccess Add-on SDK. Developers can use this guide, in conjunction with the installed Javadocs, to extend the functionality of the PingAccess server.

Intended Audience

This guide is intended for application developers and system administrators responsible for extending PingAccess. The reader should be familiar with Java software-development principles and practices. It describes the development of:

- SiteAuthenticators
- Rules

Additional Documentation

- The PingAccess Javadocs provide detailed reference information for developers. The Javadocs can be accessed with a web browser by viewing the file `<PA_install>/sdk/apidocs/index.html`.

Introduction

Introduction

The PingAccess Add-on SDK provides the following extension points:

- RuleInterceptor - An interface for developing custom Rule implementations to control authorization logic in policies.
- SiteAuthenticatorInterceptor - An interface for developing custom Site Authenticators to control how PingAccess (operating as a proxy) is able to integrate with web servers or services it is protecting.

These extension points allow users to customize certain behaviors of PingAccess to suit an organization's needs. This SDK provides the means to develop, compile, and deploy custom custom extensions to PingAccess.

If you need assistance using the SDK, visit the Ping Identity Support Center (www.pingidentity.com/support) to see how we can help you with your application.
You may also engage the Ping Identity Global Client Services team for assistance with developing customizations.

Getting Started with the SDK

Getting Started With the SDK

This section describes the directories and build components that comprise the SDK and provides instructions for setting up a development environment.

Directory Structure

The PingAccess SDK directory (<PA_install>/sdk) contains the following:

- README.md – Contains an overview of the SDK contents.
- /samples/README.md – Contains an overview of the steps necessary to build and use the samples.
- /samples/Rules – Contains a maven project with example plug-in implementations for Rules showing a wide range of functionality. You may use these examples for developing your own implementations.
- /samples/Rules/README.md – Contains the details of the Rules samples.
- /samples/SiteAuthenticator – Contains a maven project with example plug-in implementations for Site Authenticators. You may use these examples for developing your own implementations.
- /samples/SiteAuthenticator/README.md – Contains the details of the Site Authenticator samples.
- apidocs/ – Contains the SDK Javadocs. Open index.html to get started.

Prerequisites

Before you start, ensure you have the Java SDK and Apache Maven installed. The samples use Apache Maven and assume that the PingAccess SDK can be referenced as a dependency. They reference PingIdentity's public maven repository, located at:

```
http://maven.pingidentity.com/release
```

If internet access is unavailable, there are two other ways to reference the PingAccess SDK. First, once Apache Maven is installed, install the sdk into your local dependency repository by running the following command:

```
mvn install:install-file -Dfile=<PA_install>/lib/pingaccess-sdk-3.0.0.jar
-DgroupId=com.pingidentity.pingaccess -DartifactId=pingaccess-sdk -Dversion=3.0.0 -Dpackaging=jar
```

Alternatively, you can update the pingaccess-sdk dependency in your `pom.xml` to point to the local installation.

```
<dependency>
        <groupId>com.pingidentity.pingaccess</groupId>
        <artifactId>pingaccess-sdk</artifactId>
        <version>3.0.0</version>
        <scope>system</scope>
        <systemPath><PA_install>/lib/pingaccess-sdk-3.0.0.jar</systemPath>
    </dependency>
```

With either of these options, replace <PA_install> with the path to the PingAccess installation.

How to install the samples

- Before you begin, ensure you have the Java SDK and Apache Maven installed.
- Each sample type is installed separately:
    - For the Rules samples, navigate to <PA_install>/sdk/samples/Rules
    - For the Site Authenticators samples, navigate to <PA_install>/sdk/samples/SiteAuthenticator
- From the sample's directory, run the command: $ mvn install
    - This builds the samples, runs their tests, and copies the resulting jar file from the target directory to the <PA_install>/lib directory.

```
jmusgrave-MBP-2:Rules jmusgrave$ mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] Using the builder
org.apache.maven.lifecycle.internal.builder.singlethreaded.SingleThreadedBuilder with a thread count
of 1
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building PingAccess :: Sample Rules 3.0.0-RC5
[INFO] ------------------------------------------------------------------------
Downloading: http://...
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ sample-rules ---
[INFO] Using 'ISO-8859-1' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ sample-rules ---
[INFO] Compiling 7 source files to
/Users/jmusgrave/Downloads/pingaccess-3.0.0-RC5/sdk/samples/Rules/target/classes
```

```
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ sample-rules ---
[INFO] Using 'ISO-8859-1' encoding to copy filtered resources.
[INFO] Copying 4 resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ sample-rules ---
[INFO] Compiling 4 source files to
/Users/jmusgrave/Downloads/pingaccess-3.0.0-RC5/sdk/samples/Rules/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ sample-rules ---
[INFO] Surefire report directory:
/Users/jmusgrave/Downloads/pingaccess-3.0.0-RC5/sdk/samples/Rules/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.pingidentity.pa.sample.TestAllUITypesAnnotationRule
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.912 sec
Running com.pingidentity.pa.sample.TestIllustrateManyUITypesRule
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.029 sec
Running com.pingidentity.pa.sample.TestValidateRulesAreAvailable
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.002 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ sample-rules ---
[INFO] Building jar:
/Users/jmusgrave/Downloads/pingaccess-3.0.0-RC5/sdk/samples/Rules/target/sample-rules-3.0.0-RC5.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ sample-rules ---
[INFO] Installing
/Users/jmusgrave/Downloads/pingaccess-3.0.0-RC5/sdk/samples/Rules/target/sample-rules-3.0.0-RC5.jar to
/Users/jmusgrave/.m2/repository/com/pingidentity/pingaccess/sample-rules/3.0.0-RC5/sample-rules-3.0.0-RC5
Installing /Users/jmusgrave/Downloads/pingaccess-3.0.0-RC5/sdk/samples/Rules/pom.xml to
/Users/jmusgrave/.m2/repository/com/pingidentity/pingaccess/sample-rules/3.0.0-RC5/sample-rules-3.0.0-RC5
--- maven-antrun-plugin:1.7:run (default) @ sample-rules ---
[INFO] Executing tasks

main:
     [copy] Copying 1 file to /Users/jmusgrave/Downloads/pingaccess-3.0.0-RC5/lib
[INFO] Executed tasks
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6.418 s
```

```
[INFO] Finished at: 2014-07-08T16:38:30-07:00
[INFO] Final Memory: 16M/38M
[INFO] ------------------------------------------------------------------------
```

Creating your own Plugins

Creating your own Plugins

This section describes using the samples as a template for creating your own plugins.

Creating a Rule

- For details on how to create a Rule, reference the javadoc at: <PA_install>/sdk/apidocs/com/pingidentity/pa/sdk/policy/RuleInterceptor.html
- Add a Java class to /sdk/samples/Rules/src that extends com.pingidentity.pa.sdk.policy.RuleInterceptor and is annotated by com.pingidentity.pa.sdk.policy.Rule. A base class com.pingidentity.pa.sdk.policy.RuleInterceptorBase is available to simplify implementing a Rule.
- Add the class name of the new class to /sdk/samples/Rules/src/main/resources/META-INF/services/com.pingidentity.pa.sdk.policy.RuleInterceptorExecute maven install on the Rules sample pom.

Creating a Site Authenticator

- For details on how to create a Site Authenticator, reference the javadoc at: <PA_install>/sdk/apidocs/com/pingidentity/pa/sdk/siteauthenticator/SiteAuthenticator.html
- Add a Java class to/sdk/samples/SiteAuthenticator/src that extendscom.pingidentity.pa.sdk.siteauthenticator.SiteAuthenticatorInterceptor and is annotated by com.pingidentity.pa.sdk.siteauthenticator.SiteAuthenticator. A base class com.pingidentity.pa.sdk.siteauthenticator.SiteAuthenticatorInterceptorBase is available to simplify implementing a SiteAuthenticator.
- Add the class name of the new class to /sdk/samples/Rules/src/main/resources/META-INF/services/com.pingidentity.pa.sdk.siteauthenticator.SiteAuthenticator Execute maven install on the SiteAuthenticator sample pom.

Implementation Guidelines

Implementation Guidelines

The following sections provide specific programming guidance for developing custom interfaces. Note that the information is not exhaustive – consult the Javadocs to find more details about interfaces discussed here as well as additional functionality.

Logging

Use the SLF4j api for logging activities in your module. Documentation on using SLF4j is available on the SLF4j website.

Lifecycle

The plugins and the implementation of a PluginConfiguration can be instantiated for a number of reasons and at many times. For example, with a RuleInterceptor here is what happens before the RuleInterceptor is available to process user requests:

- The Rule annotation on the implementation class of the RuleInterceptor is interrogated to determine which PluginConfiguration instance will be instantiated.
- The following is performed on RuleInterceptor and PluginConfiguration. Which of these is handled first is not defined.
    - The bean will be provided to Spring for Autowiring.
    - The bean will be provided to Spring for post construction initialization. (See PostConstruct)
- PluginConfiguration.setName(String) is called.
- PA attempts to map the incoming JSON configuration to the PluginConfiguration instance.
- ConfigurablePlugin.configure(PluginConfiguration) is called.
- Validator.validate(Object, Class[]) method is invoked and provided to the RuleInterceptor.
- The instance is then made available to service end user requests, such as RequestInterceptor.handleRequest(com.pingidentity.pa.sdk.http.Exchange) and ResponseInterceptor.handleResponse(com.pingidentity.pa.sdk.http.Exchange)

Injection

Before they are put into use, Rules, SiteAuthenticators, and their defined PluginConfigurations are passed through Spring's Autowiring and initialization. To future-proof any code against changes in PingAccess, we recommend that Spring not be used as a dependency. Use the annotation javax.inject.Inject for any injection.

Classes Available for Injection

Currently, injection is available for the following classes:

- com.pingidentity.pa.sdk.util.TemplateRenderer